

JWT Release Notes 2.2.2

[2015-12-14] Released [Jira Workflow Toolbox 2.2.2](#)

New Features

- Added 2 new functions to the parser:

FUNCTION	RETURNED VALUE
textOnIssueList (<i>issue list issues</i> , <i>string text_expression</i>) : <i>string list</i>	Returns a list of strings resulting of evaluating text_expression against each of the issues in argument issues . Argument text_expression is an expression that returns a <i>string</i> , where references to field values of issues in argument issues are done with prefix ^ before field code, e.g., ^{00000} is field code for <i>Summary</i> in each of the issues in argument issues . Example: <code>textOnIssueList(subtasks(), ^{00003} = ^{00006} ? ^{00015} : null)</code> returns the list of subtasks with same user as reporter and as assignee.
mathOnIssueList (<i>issue list issues</i> , <i>number math_time_expression</i>) : <i>number list</i>	Returns a list of numbers resulting of evaluating math_time_expression against each of the issues in argument issues . Argument math_time_expression is a math/time expression, where references to field values of issues in argument issues are done with prefix ^ before field code, e.g., ^{00012} is field code for <i>Due date</i> in each of the issues in argument issues . Example: <code>mathOnIssueList(linkedIssues("is blocked by"), (^{00012} != null ? ^{00012} - ^{00009} : 0) / {HOUR})</code> returns a list of numbers with the number of days from issue creation to due date for all issues linked using "is blocked by" issue link type.

Improvements

- Post-functions " [Write field on linked issues or sub-tasks](#)", " [Write Field On Issues Returned By JQL Query](#)", " [Read fields from linked issues or sub-tasks](#)" and " [Read Field From Issues Returned By JQL Query](#)" now supports a **parsed text** or a **math/time expression** in parameter **source value**. This way:
 - In most cases the usage of *ephemeral fields* as auxiliary value container becomes unnecessary, since now you can directly input in the post-function configuration the value you want to write.
 - Other issues (linked, subtasks and JQL selected) can be updated using as source the values of fields in those issues, or expressions using the values of those fields.
- Improved usage of **null** by the parser. Now text-string and math-time expressions can return null. It can be used for unsetting fields, or as a value that will not be added to a string or number list.
- Numerous UI improvements: added field code injectors to all the features, text areas were replaced with advanced editor, revised texts in UI, translations in validations messages are shown by default when they exist, etc.
- " [Copy cascading select field](#)" post-function now supports **MultiLevel Cascading Select** and **Jira Database Values Plugin** custom fields.
- Support for reading custom fields provided by **Kepler Custom Fields Pro** plugin.
- All the javascript code has been refactorized for maintainability.

Bug Fixes

- " [Copy cascading select field](#)" post-function couldn't set "Select List" or "Radio Button" custom fields.
- Virtual field "**Attachments**" threw exception when used in a validator in "**Create Issue**" transition.
- Function "**previousValue()**" returned the value before previous one when the field was edited in transition screen.
- Issue #205** - Bug on "**numberOfRemoteIssueLinks()**" function.
- " [Parse field for extracting data](#)" didn't support delimiter definitions and format definitions longer that 255 characters.
- " [Create issue link](#)" post-function doesn't show an image correctly with Jira 7.x.